



Computadoras y programación

Grado en Ingeniería Electrónica de Comunicaciones

Luis Hernández Yáñez
Raquel Hervás Ballesteros
Virginia Francisco Gilmartín
Javier Arroyo Gallardo

Facultad de Informática
Universidad Complutense



Índice

Informática, computadoras y computación

Lenguajes de programación

Ingeniería del Software

Programación en C++: Primer contacto



Informática (Ciencia de la computación): Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.



Computadora: Máquina electrónica, analógica o digital, dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas matemáticos y lógicos mediante la ejecución de programas informáticos



Computadoras



Hardware y software

Hardware

Componentes que integran la parte material de una computadora



Software

Programas, instrucciones y reglas informáticas para ejecutar tareas en una computadora



Tipos de computadora

De propósito específico:

- ✓ Realizan una única tarea para la que fueron diseñadas.
- ✓ Poseen circuitos que realizan cálculos concretos.
- ✓ Entrada: una instancia del problema.

De propósito general:

- ✓ Pueden realizar distintas tareas.
- ✓ Entrada: las instrucciones para solventar un problema.



Programación

Programación de computadoras

- ✓ Indicar a la computadora qué es lo que tiene que hacer.

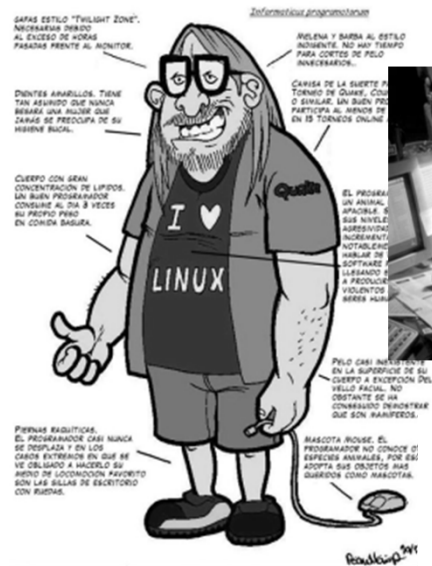
Programa

- ✓ Secuencia de instrucciones.
- ✓ Instrucciones que entiende la computadora.
- ✓ Y que persiguen un objetivo: *resolver un problema*.



Programadores

ANATOMÍA DEL PROGRAMADOR GEEK



Parque Jurásico



Trabajo en equipo Múltiples roles...

- ✓ Gestores
- ✓ Analistas
- ✓ Diseñadores
- ✓ Programadores
- ✓ Probadores
- ✓ Administradores de sistemas

...



Computación: Algoritmos y programas

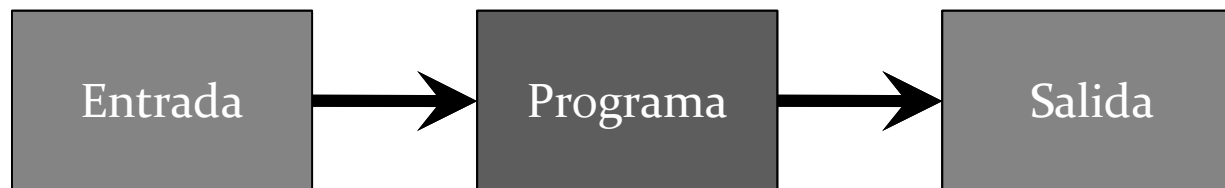
Algoritmo:

“Receta” que transforma un estado inicial en un estado final.

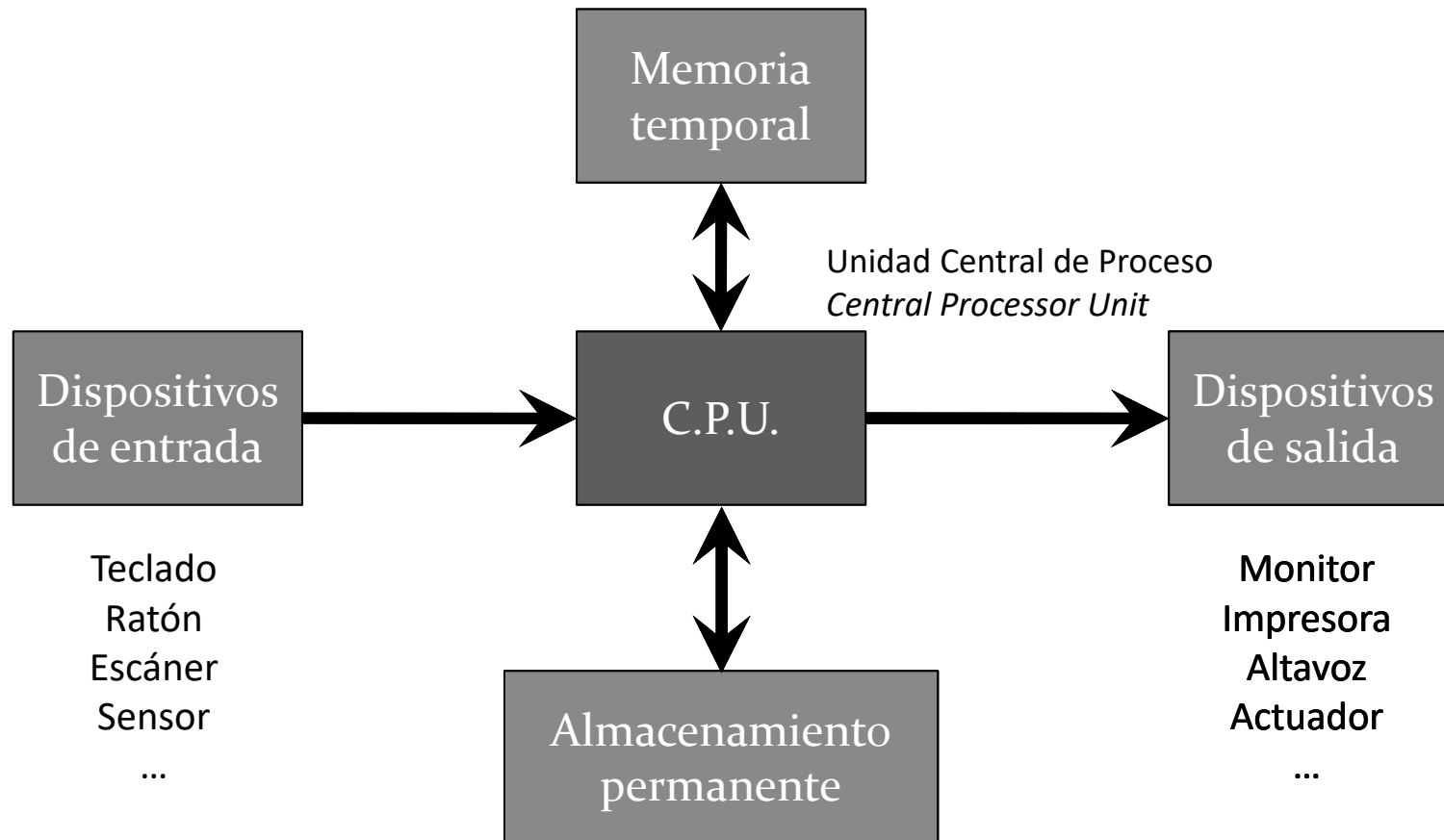


Programa:

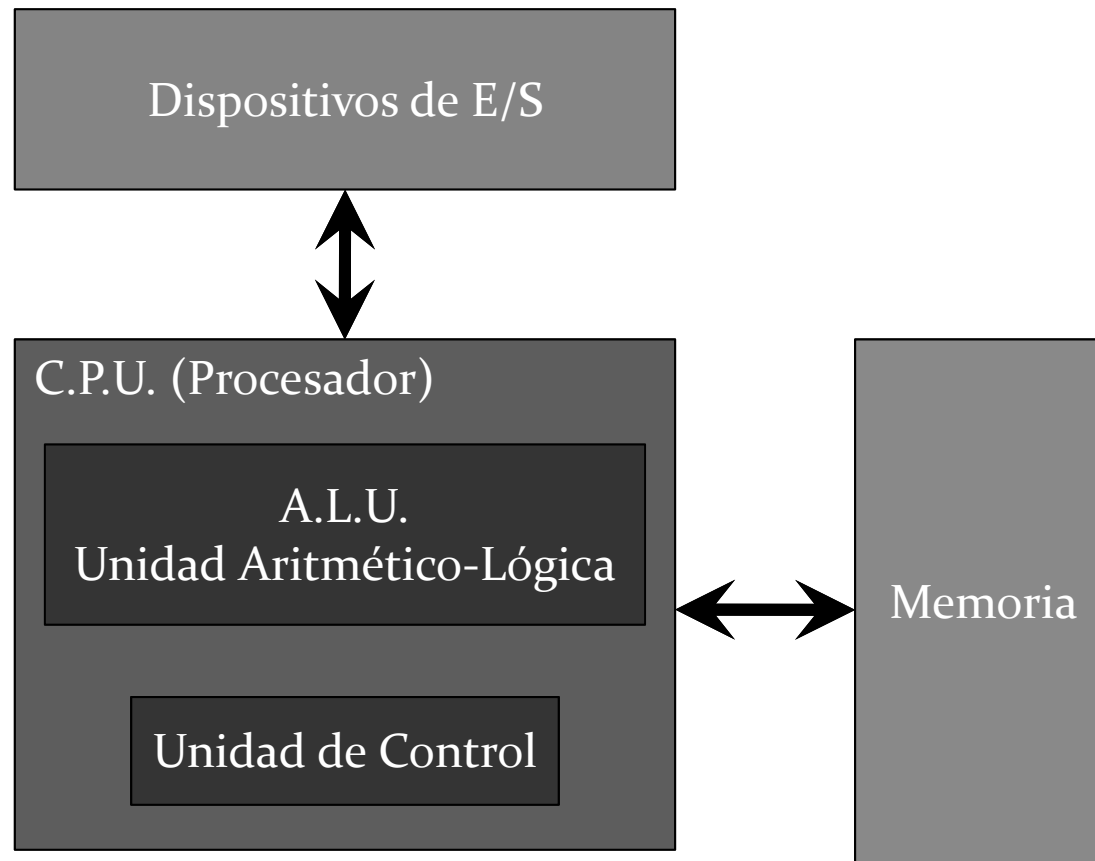
Algoritmo codificado en un lenguaje de programación concreto.



Computadoras. Esquema general

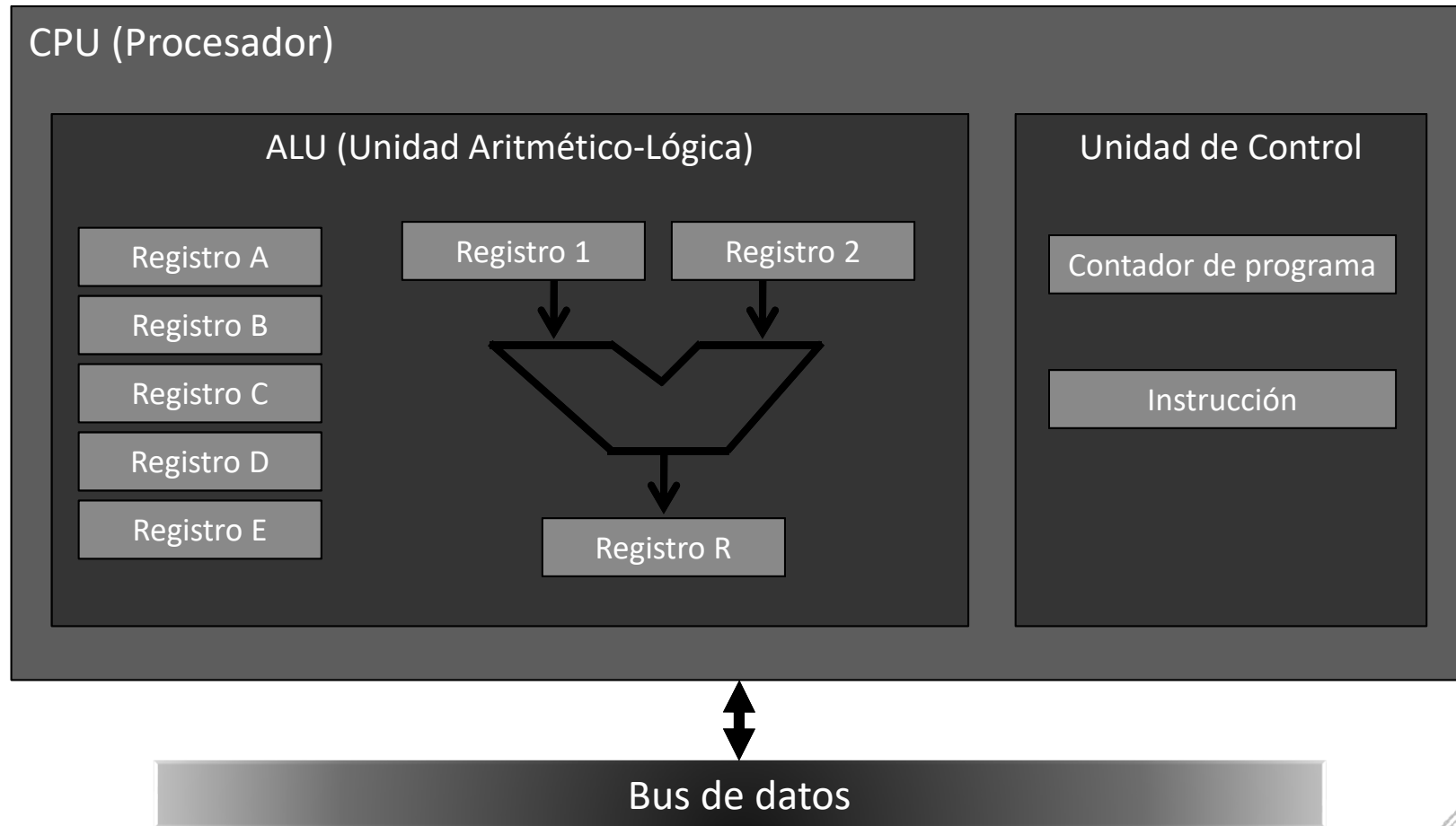


Arquitectura de Von Neumann

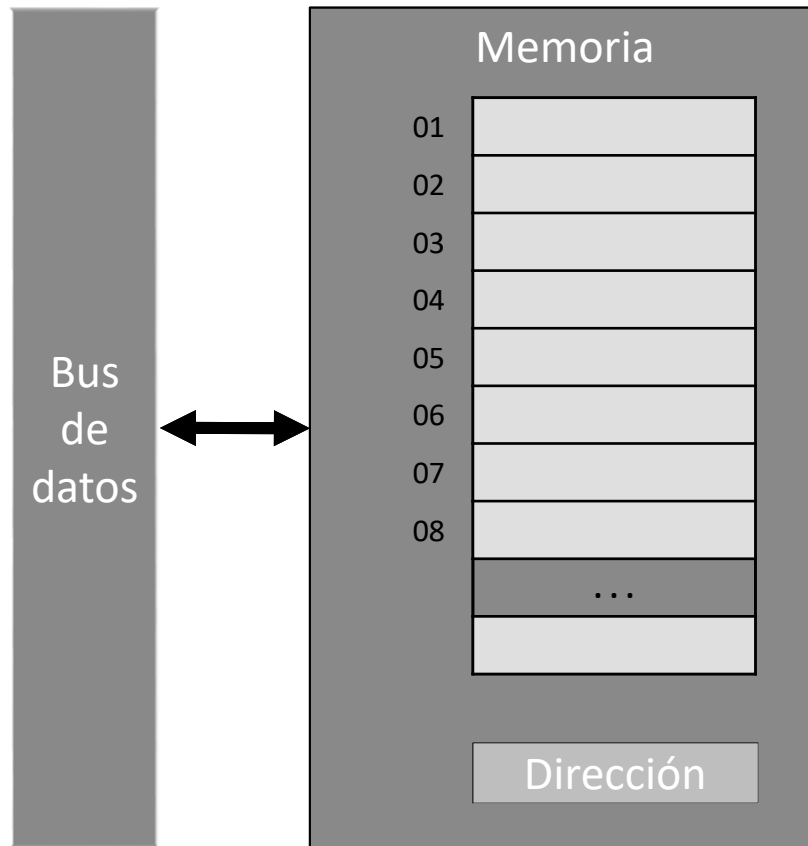


Unidad Central de Proceso (CPU)

Registros de 8 / 16 / 32 / 64 bits



Memoria



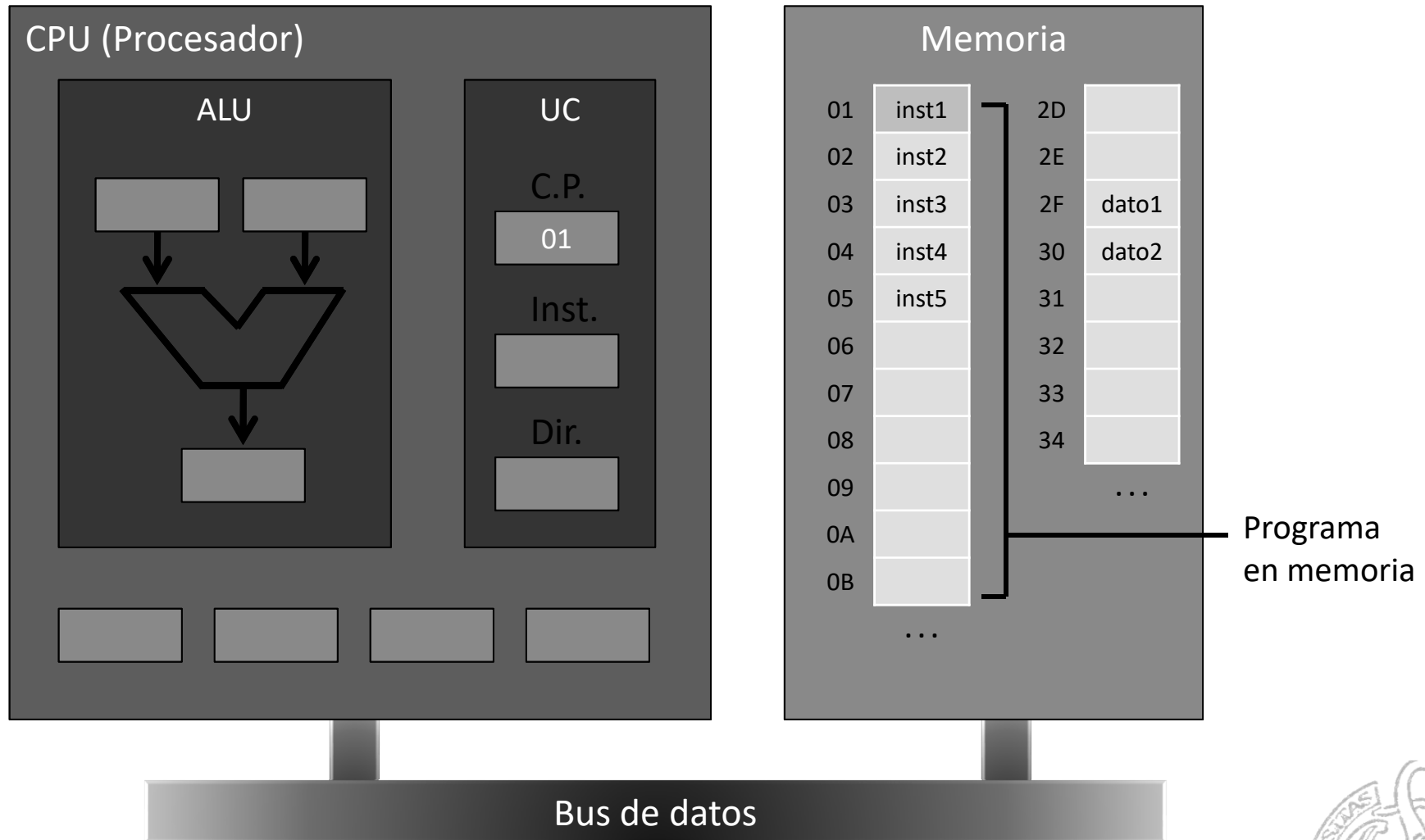
Cada celda tiene una dirección asociada.
Celdas de 8 / 16 / 32 / 64 bits
Información volátil.

1 Bit = 0 / 1
1 Byte = 8 bits = 1 carácter / nº (0-255)
1 Kilobyte (KB) = 1024 Bytes
1 Megabyte (MB) = 1024 KB
1 Gigabyte (GB) = 1024 MB
1 Terabyte (TB) = 1024 GB
1 Petabyte (PB) = 1024 TB

$$2^{10} = 1024$$



Secuenciamiento de instrucciones (I)



Secuenciamiento de instrucciones (II)

Ideas clave

- ✓ La máquina ejecuta instrucciones simples secuencialmente.
- ✓ Algunas instrucciones contienen tests que comprueban una condición y, si es cierta, cambian el Contador de Programa (control de flujo).
- ✓ Preparamos nuestros programas usando un conjunto de operaciones primitivas simples para realizar tareas complejas.



El lenguaje máquina (I)

- ✓ Los procesadores trabajan con ceros y unos.
- ✓ La unidad de memoria básica: Byte (8 bits).
2 dígitos hexadecimales: 01011011 → 0101 1011 → 5B
- ✓ Lenguaje máquina: Códigos hexadecimales que representan instrucciones, registros de la CPU, direcciones de memoria o datos.
- ✓ Programa: secuencia de códigos que instruyen a la CPU.

Instrucción *Significado*

| | | |
|----|----|--|
| A0 | 2F | <i>Acceder a la posición de memoria 2F</i> |
| 3E | 01 | <i>Copiar el dato en el registro 1 de la ALU</i> |
| A0 | 30 | <i>Acceder a la posición de memoria 30</i> |
| 3E | 02 | <i>Copiar el dato en el registro 2 de la ALU</i> |
| 1D | | <i>Sumar</i> |
| B3 | 31 | <i>Guardar el resultado en memoria (31)</i> |



El lenguaje máquina (II)

- ✓ Lenguaje de programación de bajo nivel. Nulo nivel de abstracción: los códigos contienen los ceros y unos que gobiernan directamente los circuitos de la CPU.
- ✓ Totalmente dependiente de la máquina. Cada familia de procesadores usa sus propios códigos, distintos de los de otras familias.
- ✓ La programación en lenguaje máquina es muy tediosa.



El lenguaje ensamblador

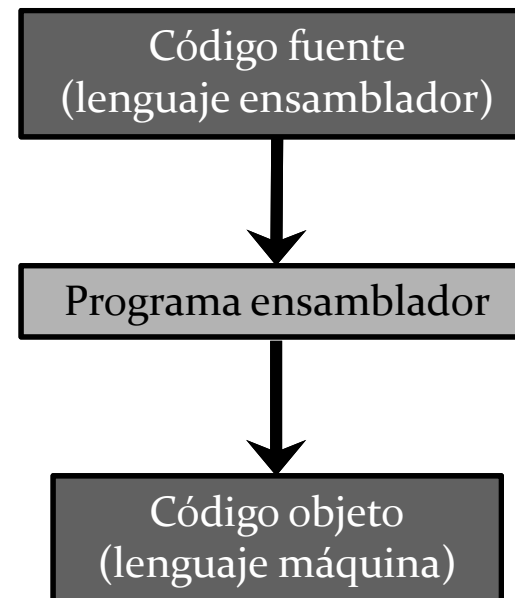
- ✓ Nemotécnicos para los códigos hexadecimales que representan instrucciones:

A0 → READ 3E → REG 1D → ADD ...

- ✓ Mayor legibilidad:

READ 2F
REG 01
READ 30
REG 02
ADD
WRITE 31

- ✓ Lenguaje de nivel medio.



Lenguajes de programación de alto nivel (I)

- ✓ Más cercanos al lenguaje natural y al lenguaje matemático.
`resultado = dato1 + dato2;`
- ✓ Mayor legibilidad, mayor facilidad de codificación.
- ✓ Estructuración de datos / abstracción procedimental

FORTRAN Python Prolog C#
C Pascal Cobol Lisp Ruby
BASIC Smalltalk Haskell Ada
Simula Java Eiffel C++
...

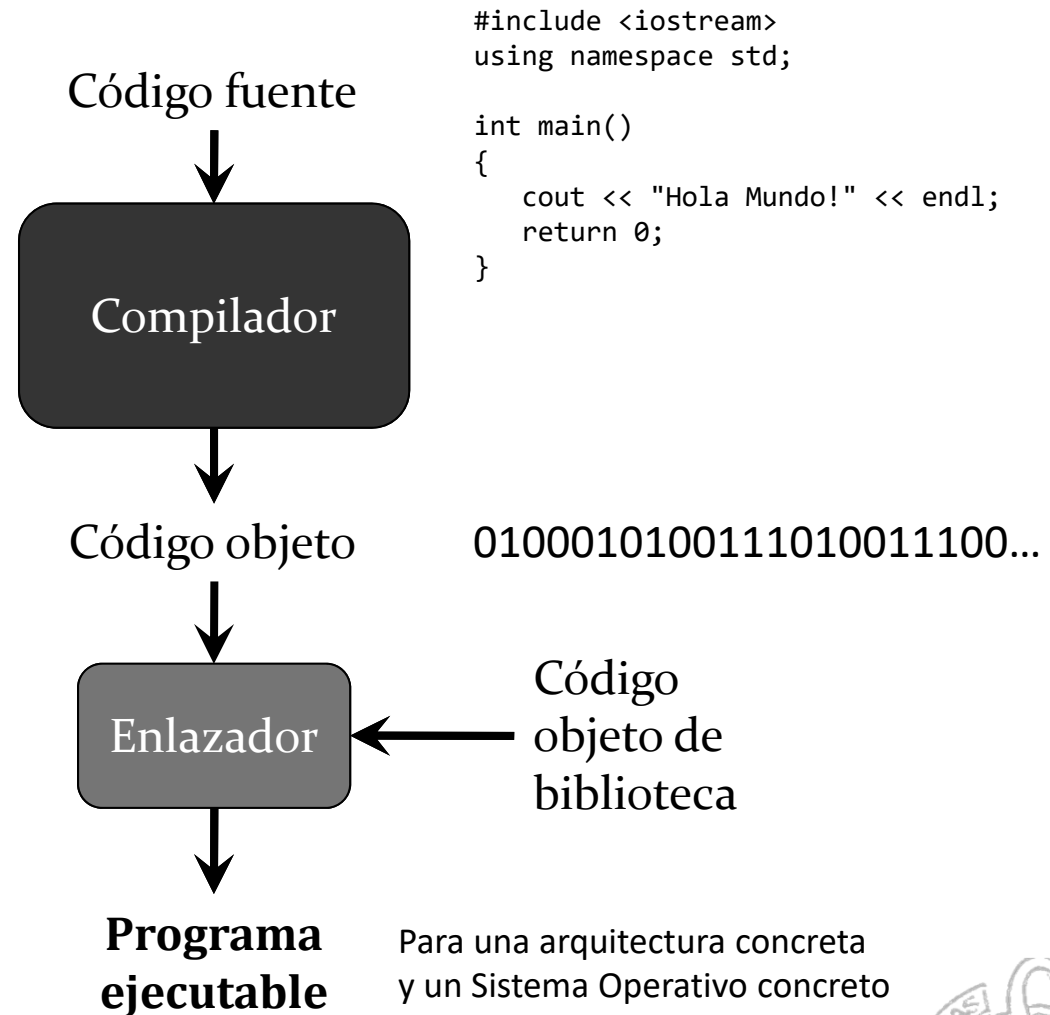


Lenguajes de programación de alto nivel (II)

Traducción

Compiladores:
Compilan el código
fuente a código objeto.

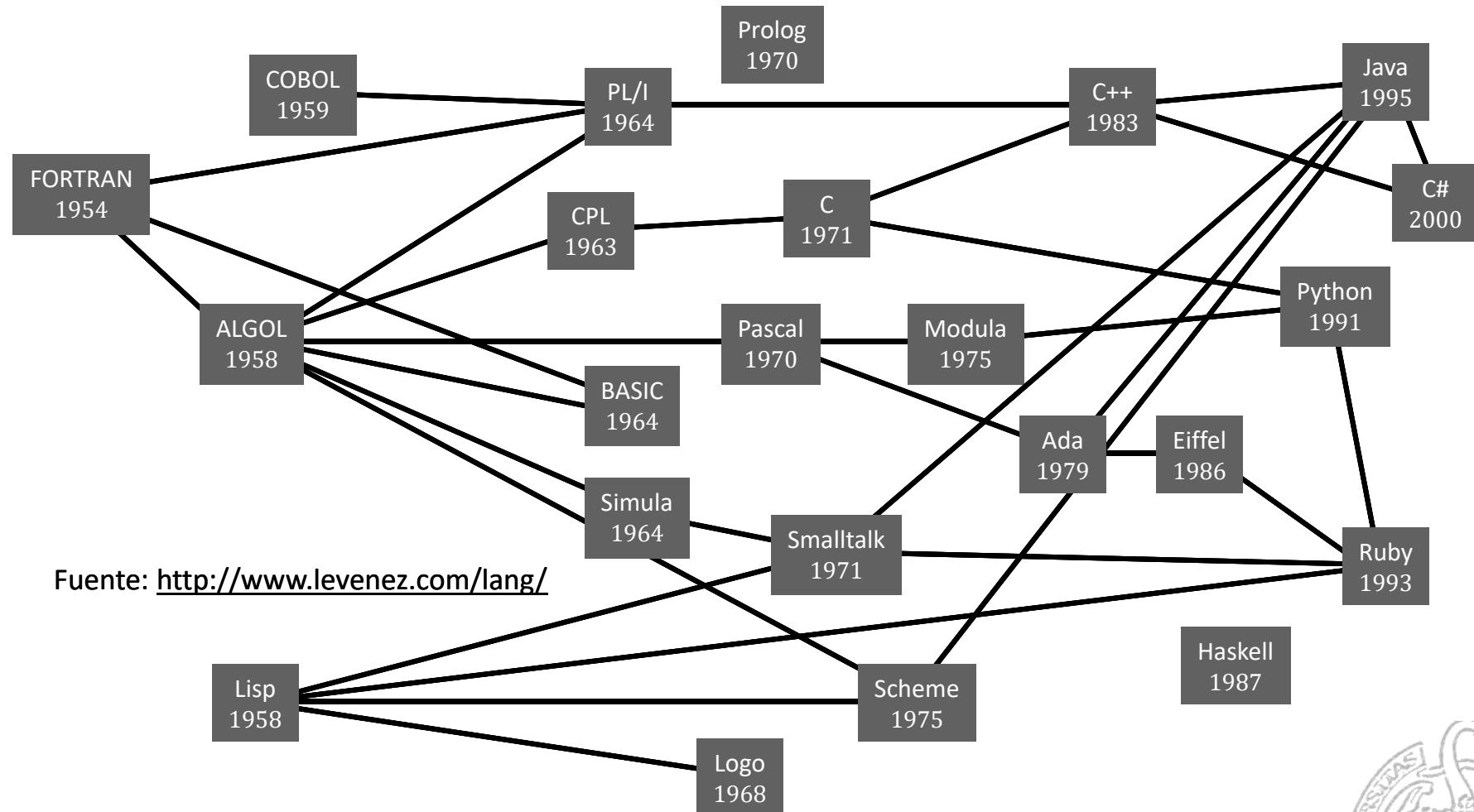
Enlazadores:
Enlazan distintos
códigos objeto en
programas completos.



Lenguajes de programación de alto nivel (III)

Genealogía de lenguajes

Versiones / Estándares



Fuente: <http://www.levenez.com/lang/>



El lenguaje de programación C++

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hola Mundo!" << endl; // Muestra Hola Mundo!

    return 0;
}
```

Hola Mundo!



Elementos del lenguaje

Instrucciones

Datos: literales, variables, tipos

Subprogramas (funciones)

Comentarios

Directivas

...

```
#include <iostream>  
using namespace std;
```

Directiva

Subprograma

```
int main()  
{
```

Dato

Instrucción

```
cout << "Hola Mundo!" << endl;  
// Muestra Hola Mundo!
```

Comentario

Instrucción

```
return 0;  
}
```

Dato



¿Por qué C++?

- ✓ Para programar necesitamos un lenguaje.
- ✓ C++ es el lenguaje más ampliamente utilizado en las áreas de ingeniería:

<http://www.research.att.com/~bs/applications.html>

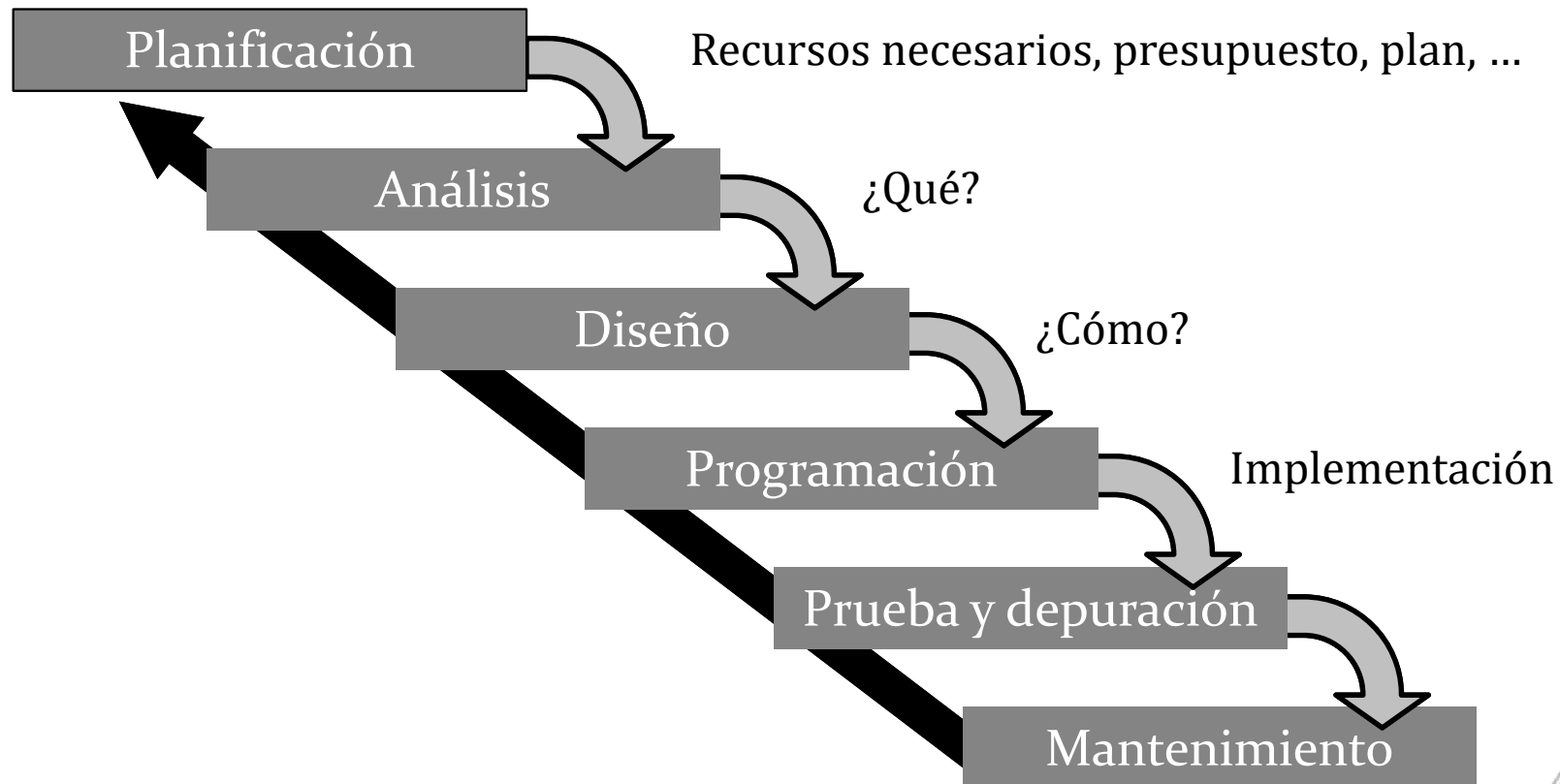
- ✓ C++ está bien definido por un estándar.
- ✓ C++ está disponible para casi cualquier computadora.
- ✓ Los conceptos son trasladables a otros lenguajes.



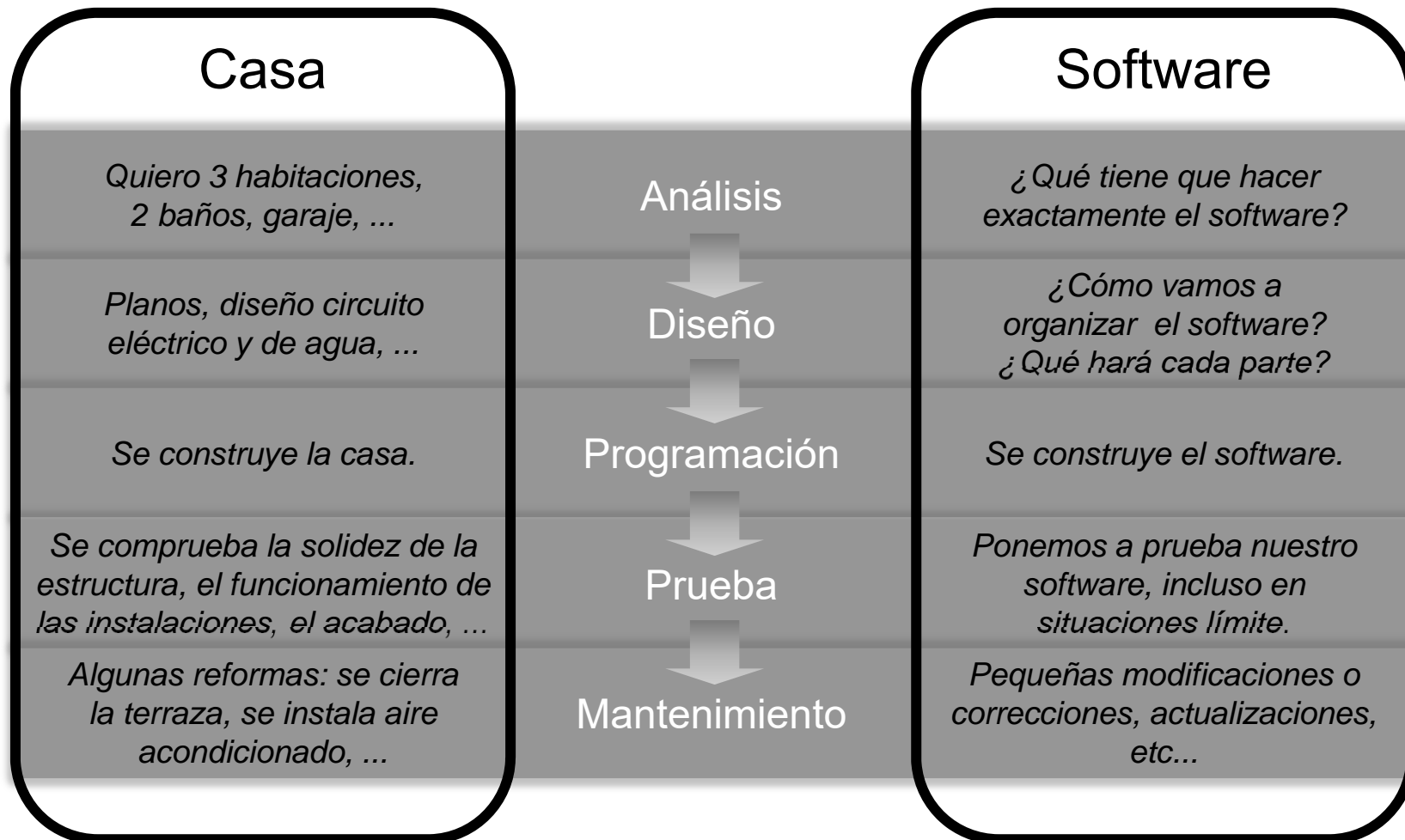
La Ingeniería del Software (I)

La programación es sólo una etapa del proceso de desarrollo.

El modelo de desarrollo “en cascada”:



La Ingeniería del Software (II)



Análisis

- ✓ Actividad en la que se analizan y clarifican los diferentes aspectos del problema que debe ser resuelto por la aplicación, con el fin de establecer claramente qué debe ser construido
- ✓ El resultado es, normalmente, un documento de requisitos software que especifica claramente las funcionalidades de la aplicación
- ✓ Funcionalidad = lo que tiene que hacerse
- ✓ Funcionalidad \neq cómo tiene que hacerse



Diseño

- ✓ Actividad en la que se decide la organización y la estructura de una aplicación que satisfaga los diferentes requisitos establecidos en la fase de análisis
- ✓ El resultado es uno (o varios) documentos de diseño que especifican claramente cómo construir la aplicación
- ✓ Mientras que el análisis se ocupa de qué hay que hacer, el diseño se ocupa de cómo hacerlo



Implementación

- ✓ Actividad en la que se construye la aplicación utilizando un lenguaje de programación concreto, y siguiendo, para ello, las directrices marcadas por los documentos de diseño
- ✓ Si las actividades anteriores han sido realizadas correctamente la fase de implementación es, en gran medida, mecánica



Pruebas

- ✓ Actividad en la que se asegura que la aplicación construida satisface los requisitos del usuario
- ✓ Dos pasos diferenciados
 - Verificación: ¿Se ajusta la aplicación construida a los requisitos establecidos?
 - Validación: ¿Resuelve la aplicación el problema que realmente tenía el usuario?



Mantenimiento

- ✓ Actividad en la que la aplicación se modifica para satisfacer cambios o ampliaciones en los requisitos del usuario, corregir errores, etc.

✓ ¡Es la actividad más costosa en el desarrollo de software!



C++: Un mejor C

La base de C++: El lenguaje C

- ✓ Lenguaje creado por Dennis M. Ritchie en 1972.
- ✓ Lenguaje orientado a la construcción de sistemas operativos (UNIX).
- ✓ Lenguaje de nivel medio:
 - Estructuras típicas de los lenguajes de alto nivel.
 - Construcciones para control a nivel de máquina.
- ✓ Lenguaje sencillo.
- ✓ Lenguaje estructurado.
- ✓ Compartimentalización de código (funciones) y datos (ámbitos).
- ✓ Componente estructural básico: la función.
- ✓ Programación modular.
- ✓ Distingue entre mayúsculas y minúsculas.
- ✓ Palabras reservadas (o clave): en minúsculas.

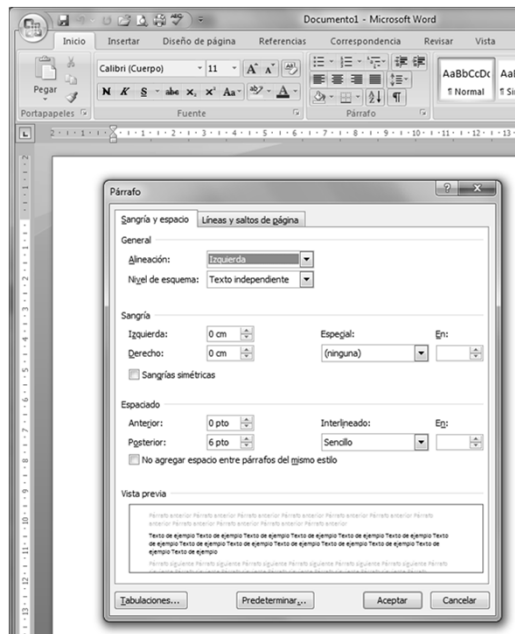


El dispositivo de salida (I)

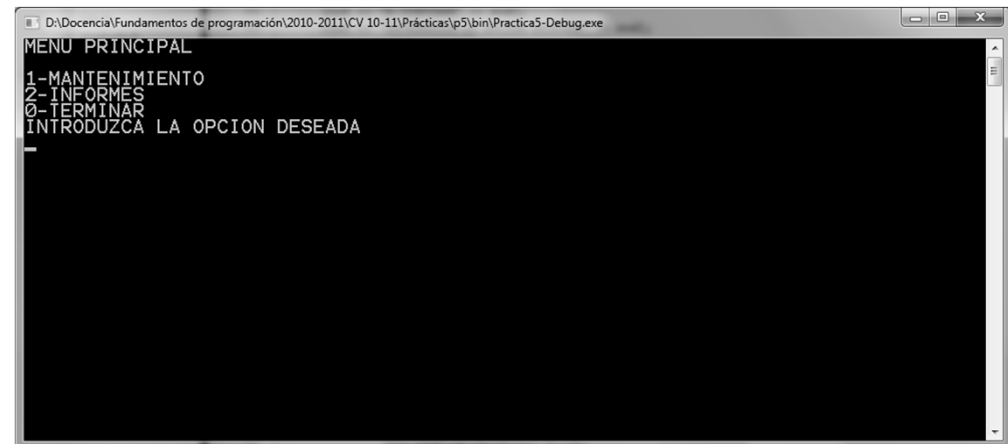
Pantalla en modo texto

En este primer curso, toda la salida se va a realizar en pantallas en modo texto. ➔ Líneas de 80 caracteres. Se pueden mostrar textos en ellas.

Aplicación con interfaz gráfica



Aplicación en modo texto



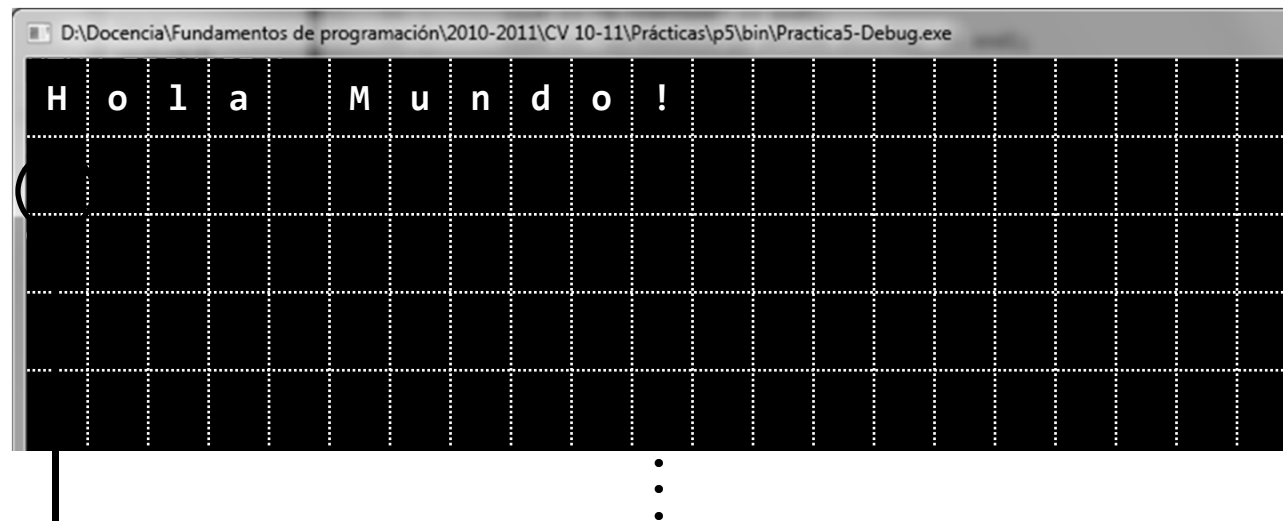
El dispositivo de salida (II)

Ventanas de consola o terminal

Sistema operativo: con interfaz gráfica de usuario (GUI).

Las aplicaciones en modo texto se ejecutan dentro de ventanas:

- ✓ Windows: ventanas de consola (*Símbolo del sistema*).
- ✓ Linux: ventanas de terminal.



Cursor parpadeante: Donde se colocará el siguiente carácter.



Un primer programa en C++ (I)

Hola Mundo!

Un programa que muestra un saludo en la pantalla:

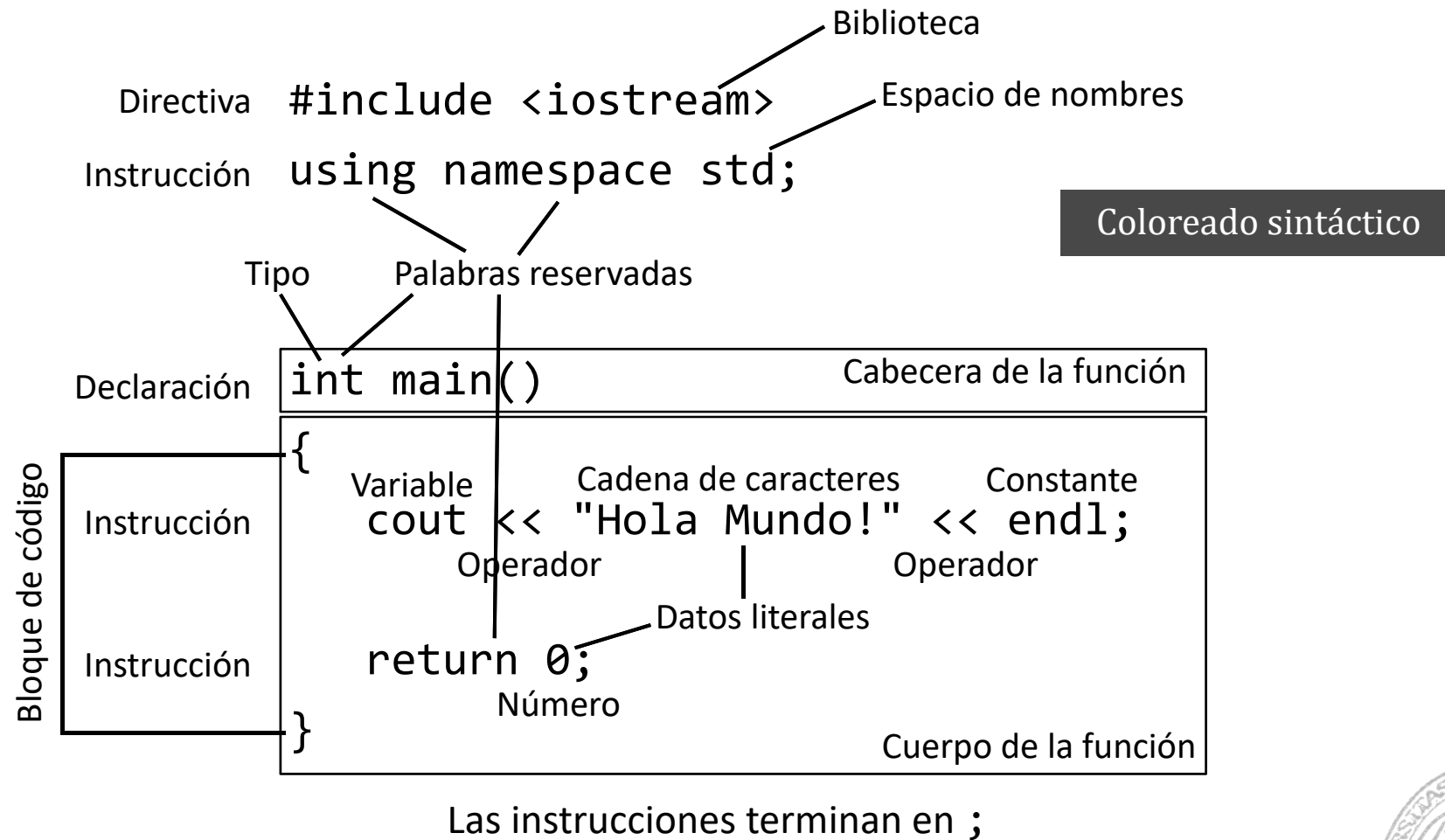
```
#include <iostream>
using namespace std;

int main()    // main() es donde empieza la ejecución
{
    cout << "Hola Mundo!" << endl; // Mostrar Hola Mundo!
    return 0;
}
```

Una secuencia de instrucciones que damos a la computadora para que las ejecute.



Un primer programa en C++ (II)



Un primer programa en C++ (III)

Hola Mundo!

Casi todo es “infraestructura”.

Sólo

```
cout << "Hola Mundo!" << endl
```

hace algo directamente.

No basta con “hacer las cosas”. Debemos tener cuidado en que las cosas se hagan de forma elegante, correcta y facilitando la creación de más/otro software.

¡El estilo importa!



Un primer programa en C++ (III)

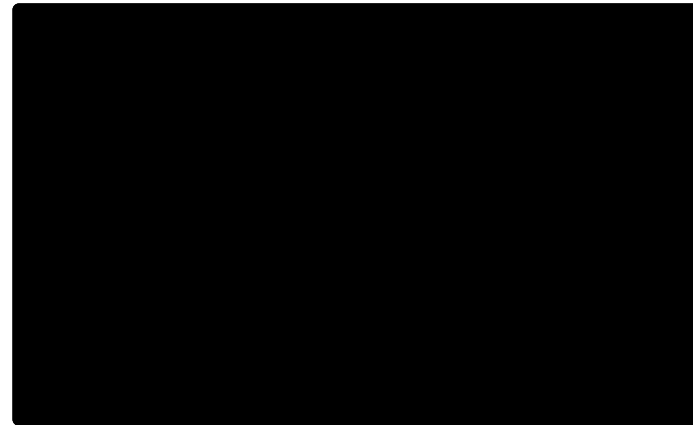
Visualización en la pantalla:

```
cout << "Hola Mundo!" << endl;
```



```
cout << "Hola Mundo!" << endl;
```

`endl` → *end line*



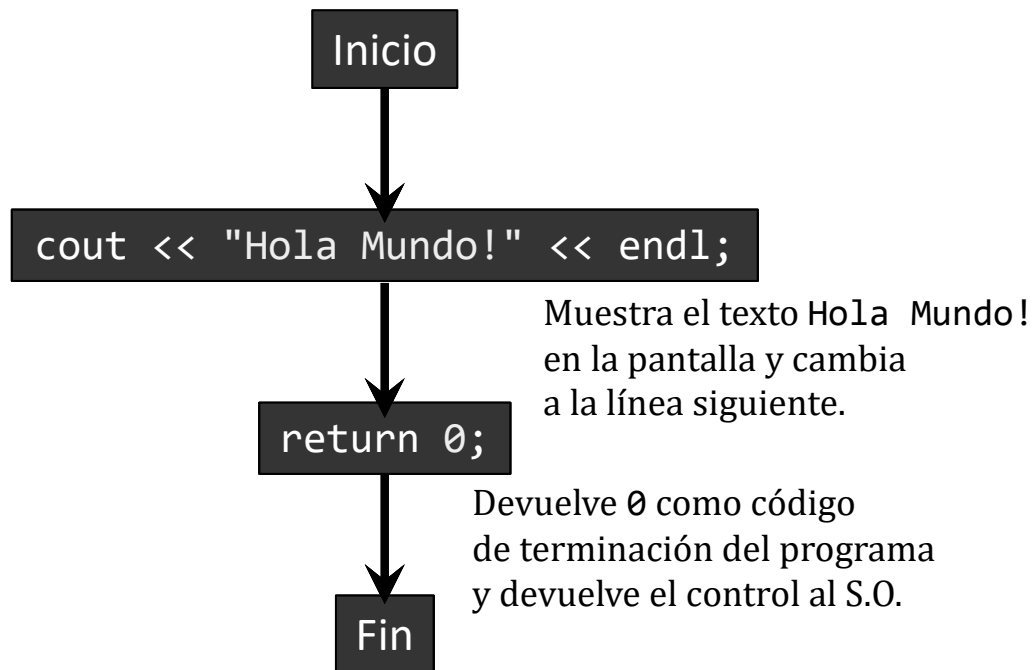
Un primer programa en C++: ejecución

¿Qué hace el programa?

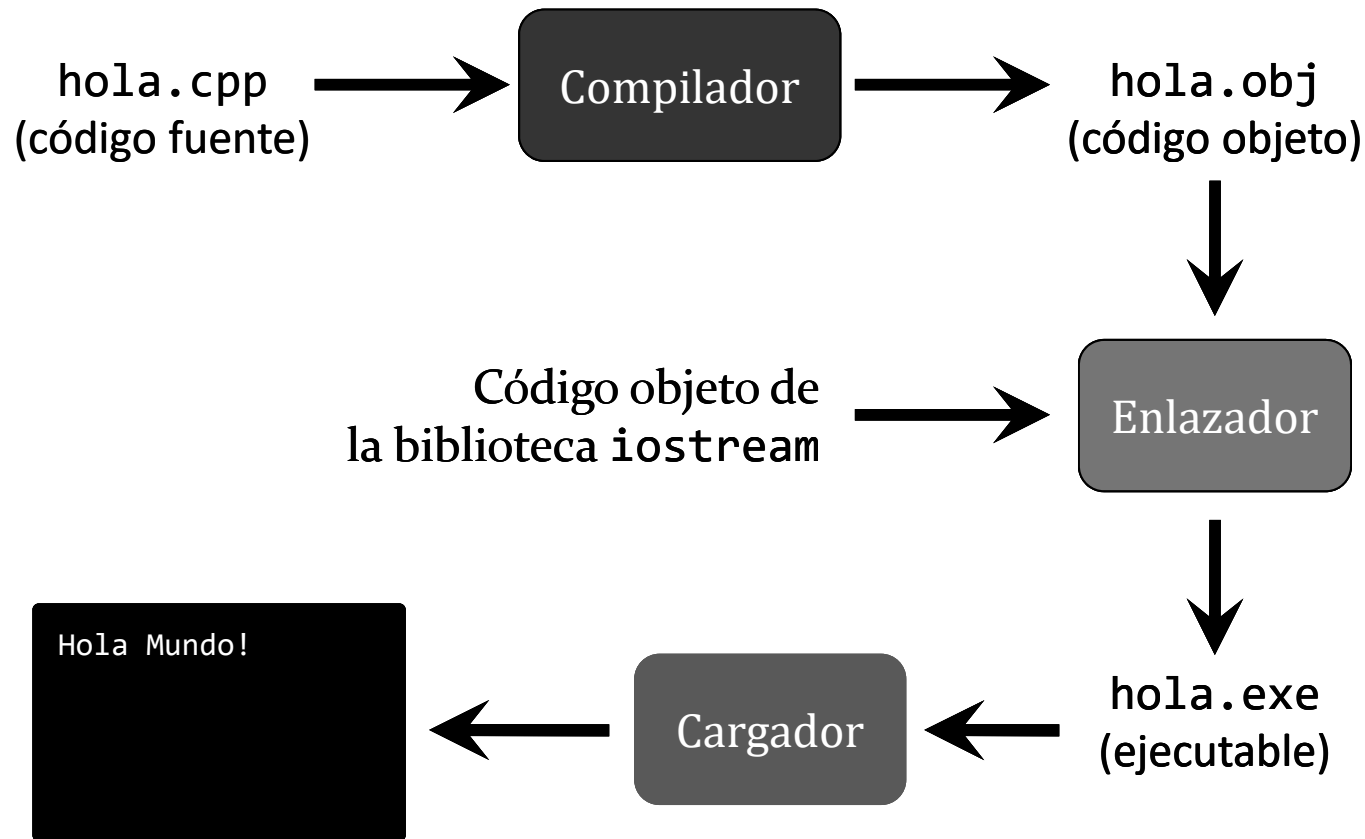
- ✓ La ejecución del programa siempre empieza en `main()`.
- ✓ Se ejecutan las instrucciones en secuencia de principio a fin (algunas instrucciones provocan saltos en la secuencia).

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hola Mundo!" << endl;
    return 0;
}
```



Compilación, enlace y ejecución

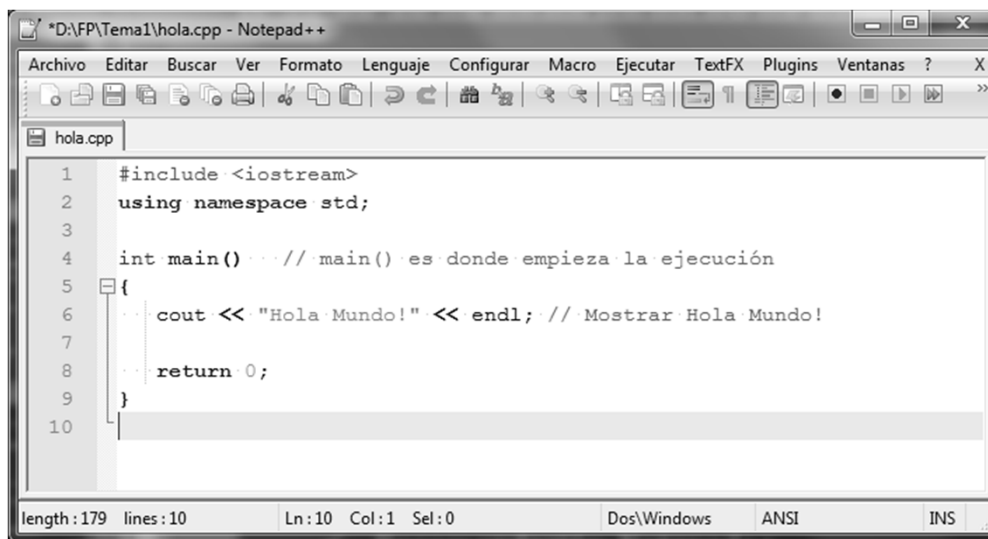


A menudo el compilador y el enlazador están integrados.



Herramientas de desarrollo. Editor

- ✓ Para escribir y modificar el código del programa.
- ✓ Bloc de notas, Wordpad, Word, ...
(Texto simple, sin formatos.)
- ✓ Editores específicos: coloreado sintáctico.
- ✓ Por ejemplo: Notepad++



The screenshot shows the Notepad++ application window with the file 'hola.cpp' open. The code is as follows:

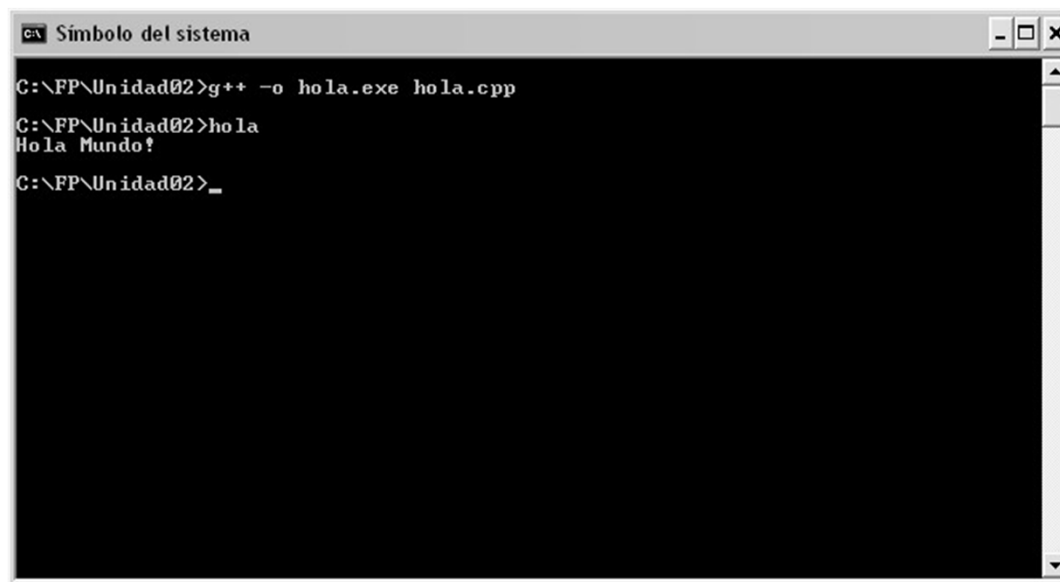
```
1 #include <iostream>
2 using namespace std;
3
4 int main() // main() es donde empieza la ejecución
5 {
6     cout << "Hola Mundo!" << endl; // Mostrar Hola Mundo!
7
8     return 0;
9 }
10
```

The status bar at the bottom indicates 'length: 179 lines: 10', 'Ln: 10 Col: 1 Sel: 0', and the encoding is 'ANSI'.



Herramientas de desarrollo. Compilador

- ✓ Para traducir (compilar) el código del programa
- ✓ Importante: C++ estándar
- ✓ Recomendación: GNU G++

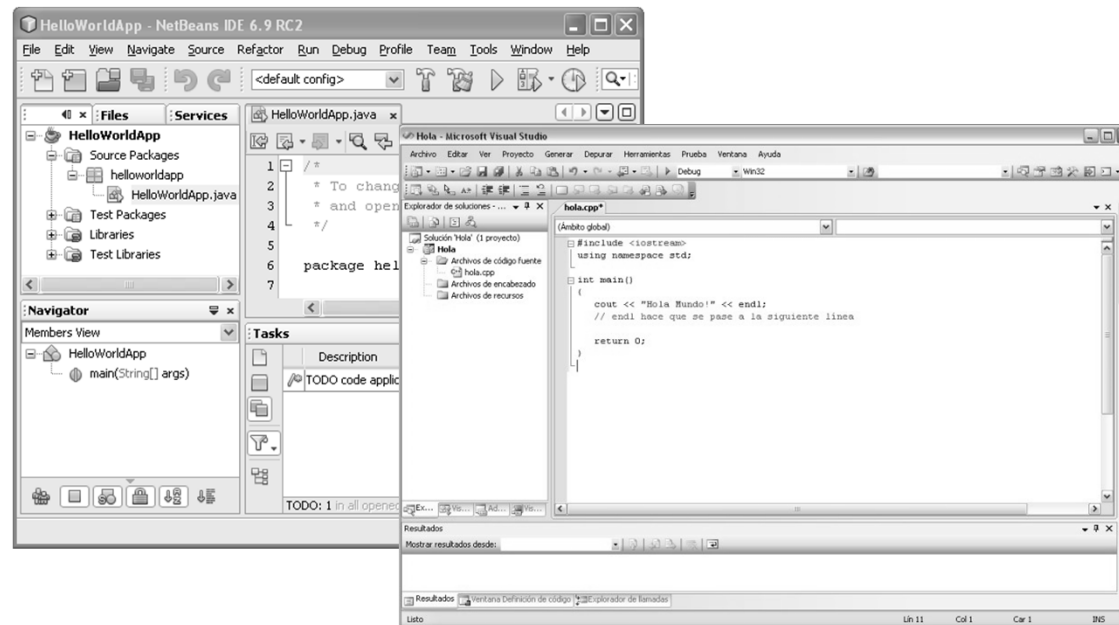


```
C:\FP\Unidad02>g++ -o hola.exe hola.cpp
C:\FP\Unidad02>hola
Hola Mundo!
C:\FP\Unidad02>_
```



Herramientas de desarrollo. Entornos de desarrollo

- ✓ Para editar, compilar y probar el código del programa.
- ✓ Múltiples tipos de aplicaciones y lenguajes.
- ✓ Recomendaciones:
 - Microsoft Visual Studio / Microsoft Visual C++ Express
 - Netbeans
 - Codeblocks






Acerca de *Creative Commons*

Licencia CC (Creative Commons)

Este tipo de licencias ofrecen algunos derechos a terceras personas bajo ciertas condiciones.

Este documento tiene establecidas las siguientes:

-  Reconocimiento (*Attribution*):
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.
-  No comercial (*Non commercial*):
La explotación de la obra queda limitada a usos no comerciales.
-  Compartir igual (*Share alike*):
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Material original elaborado por Luis Hernández Yáñez, con modificaciones de Raquel Hervás Ballesteros , Virginia Francisco Gilmartín y Javier Arroyo Gallardo.

